AIAA

CGNS-BASED DATA MODEL FOR TURBINE BLADE OPTIMIZATION

Horea Iepan
François Guibault
Marie-Gabrielle Vallet
Département de génie informatique
École Polytechnique de Montréal
Case postale 6079, succursale Centre-ville
Montréal (Québec), H3C 3A7, CANADA
Robert Magnan
Institut de recherche d'Hydro-Québec (IREQ)
1800, boul. Lionel-Boulet,
Varennes (Québec), J3X 1S1, CANADA.

CGNS-BASED DATA MODEL FOR TURBINE BLADE OPTIMIZATION

Horea Iepan*
François Guibault†
Marie-Gabrielle Vallet‡
Département de génie informatique
École Polytechnique de Montréal
Case postale 6079, succursale Centre-ville
Montréal (Québec), H3C 3A7, CANADA
Robert Magnan§
Institut de recherche d'Hydro-Québec (IREQ)
1800, boul. Lionel-Boulet,
Varennes (Québec), J3X 1S1, CANADA.

Abstract

This paper presents a CFD data model specifically designed to address the needs of the OPALE project, whose goal is to automate the design optimization of hydraulic turbine blades. The purpose of this CFD model, based on the CGNS system, is to enable the integration of commercial and in-house analysis and design applications, and to allow the management and exchange of data in order to provide a support to MDO.

The paper first presents the OPALE CFD data model structure based on the CGNS system. Some constraints in the use of CGNS are proposed in the context of the OPALE project, some interpretation problems concerning the CGNS system are resolved and the specific choices made are justified. The result is a complete CFD data model directly based on the CGNS standard specifications.

The second part of the paper presents a flexible method, based on XML, proposed to drive the filtering process of the CGNS data files produced by commercials CFD solvers, with the aim of bringing them in the OPALE standardized CGNS form.

1. Introduction

Recent evolution in processes development of a product is characterized by constantly growing complexity and the need for putting up with increasingly short design cycles. Engineering practices face major changes at all levels of production and in all industrial fields. Although these changes are felt in all fields of industry and, particularly in the automobile sector, their impact is much more obvious in those areas of engineering where products are made in limited series, and where the design represents a large fraction of the total cost of products. This is the case for the aircraft and for hydraulic turbo-machinery industry. Changes in traditional engineering practices have initiated a strong efforts at the research and development (R&D) level, in topics related to automation and multi-disciplinary design optimization. These newly developed design approaches aim to be incorporated in larger, more automated engineering systems, which must lay out effective data exchange capacities to allow their various components to quickly share information without losses.

In the field of CFD, the numerical analysis process can be represented in a relatively uniform manner. The activities which make up this numerical analysis process model include: construction of the geometrical model, generation of the grid, numerical resolution of the equations and analysis and comparison of the solutions.

The various entities which form such complete numerical analysis systems are often heterogeneous pieces of software developed both by commercial vendors and in-house software developers answering to custom needs. In most instances, software developers try to give their programs a certain degree of standardization, by equipping them with the capacity to use several input and output formats. But, due to differences introduced by each developer at implementation level, compatibility is often not satisfied.

This paper presents, the result of a study of a CFD data representation model, developed within the framework

*Graduate Student
†Associate Professor
‡Research Professional
§Research Engineer

of the OPALE project. The OPALE data model, based on the CGNS standard, satisfies the specific needs of an automated optimization system by facilitating the application of multidisciplinary optimization (MDO) practices. In other words, it makes possible to consider the analysis and design activities in an iterative cycle.

The paper documents the constraints on the use of CGNS in the framework of the OPALE project, the interpretation problems of the CGNS standard and the particular choices that were made. Commercials CFD solvers now provide the facility to use the CGNS file format, but the data structure and the content of the information produced by each solver are not completely conforming with the CGNS specifications. The paper also presents a flexible and advanced method, based on XML, to lead the filtering of CGNS files produced by various commercial CFD solvers and to bring them in the standard OPALE form.

2. Data Exchange Strategies

As a first step in the development of a data model for the OPALE project, several published standards and exchange approaches where studied. This study specifically aimed to identify well suited representation models that could cope with the CFD data representation problem itself, while proposing acceptable solutions for representing CAD data and maintaining the relationship between CFD solutions and CAD. This section presents a summary of some of the main technologies identified, and how these can collectively be put to work to form a loosely coupled, yet integrated, system.

Translators

Several approaches are used today with an aim of circumventing compatibility problems at the level of data exchanges and sharing between applications.

The most often used approach to circumvent this problem is based on translators. The output of each process passes through a translator that, by translating the input format in the desired output format, ensures the bond with the following process (see Fig. 1). A translation process in this context is rather an adaptation process from a format to another, rather than a translation. A translation implies an information transfer from a representation system to another without loss of information, which implies that even if the data representation modes are different, the data must be the same. This is not the case for most representation formats used today.

In most cases, even through the recourse to carefully designed translators, the intrinsic problem of too many incompatible formats remains. The approach based on translators presents several disadvantages : the generation of redundant data (the same data in several formats), useless use of CPU resources, wasted time of development (n formats require $n \times (n-1)$ translators), the possibility of data losses due to the translation and the loss of formats reliability. The problem is far from being solved, and solutions based on translators are more and more questioned.

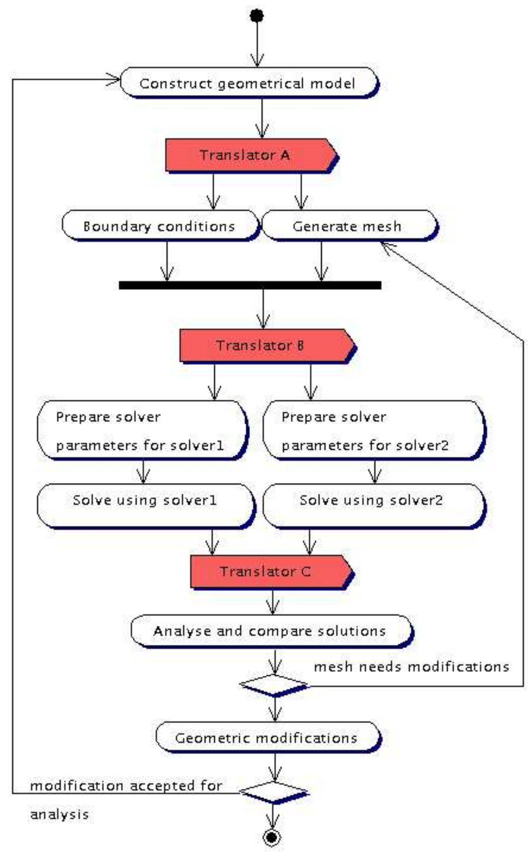

Fig. 1: The analysis process

A better, yet imperfect solution thus seems to promote the adoption of true data standards, that form core data contents that can be manipulated by growing numbers of tools and applications.

CAD data representation

In the field of CAD data representation, two main standards coexist that need to be considered, namely IGES and STEP. IGES, which was proposed several years ago, is now being phased out to the profit of STEP. Several companies, such as STEP Tools ([1]) and ProSTEP ([2]) develop suites of tools able to make the translation from the IGES format to the newer STEP format.

An alternative to this approach, suggested and implemented by Zhang et al. ([3]) is based on translators distributed on the WEB. The translation applications are distributed on Internet, and applications use them through a Java interface. The site gobbles the file by translating it. Once the translation operation is finished, the new file in the desired format will be ready to be downloaded from the site. Bhandarkar and al.([4]) present the development of an IGES to STEP AP202 translator. Basu et Kumar([5]) developed an IGES based translator for finite element analysis which extracts the grid entities - nodes and elements - from an IGES file, and convert it into STEP entities for finite elements.

The STEP standard defines abstract concepts using a data specification language named EXPRESS. These concepts are structured among application domains named *Application Protocols (AP)*, that define relevant parts of the standard that can be separately handled by tools for specific applications. The standard also defines a persistent file format to exchange data among applications.

In a more limited context, T. L.Benyo([6]) describes an approach for creating an environment able to access the geometrical data and to restore them to CFD applications directly from a CAD file. The approach uses two components *Project Integration Architecture* (PIA) and *Computational Analysis Programming Interface* (CAPRI). PIA is an object-oriented architecture that captures and integrates the elements relating to aerospace research activities. CAPRI is a programming interface created in order to acquire geometries directly from CAD files. CAPRI is in fact a collection of libraries specific to each CAD systems vendor. Until now, CAPRI has libraries to support the formats of: Unigraphics, ProEngineer, CATIA, FELISA, Computervision's CADDS and SDRC's I-DEAS.

In this context, STEP appears as the most widely spread and stable standard for CAD exchange.

CGNS - CFD General Notation System

On the front of CFD data representation, an alternative which is gaining in popularity consists in using a general representation format to keep all informations needed by the numerical analysis process. This time, the processes use a data format as a database that they can both query and update, rather than a data file either used as input or output. The CGNS ([7]) system consists of a collection of conventions, and conforming software, for storage and management of CFD data. It goal is to facilitates the CFD data exchange between CFD based analysis applications. CGNS was conceived to promote *plug-and-play* CFD.

The CGNS proposal defines concepts at an abstract level, named the *Standard Interface Data Specification (SIDS)*, as well as an application programming interface named the Mid-Level Library (MLL) and a tree structured file format name the Advanced Data Format (ADF). The CGNS standard is not conceived to store geometries([8]). The CGNS designers decided not to use a specific format of storage for the geometry but rather to use bonds toward existing CAD formats like IGES, STEP, Pro-Engineer, etc. Therefore, CGNS defines the association grid-geometry while referring to CAD entities directly in their files.

XML eXtensible Markup Language

XML ([9]) is a metalanguage of tags which was created to describe data. At the time it was created, the goal was to propose a generic language usable on the Web, but rapidly, due to its general information structure, it became an attraction point for handling among other things geometrical data, where standards like IGES and STEP reigned. XML can be defined as a technology which can create connections between structures which are comprehensible by the humans and interpretable by machines.

The tags which make up the language are not predefined, giving the user the possibility of choosing them. That confers to XML its great generality. The XML language has a mechanism enabling the description of data types by imposing restrictions at the structural and syntactic levels. To describe data types, it uses data dictionaries (Standard Document Definition-DTD) or schemas (SCHEMA). Most of the XML documents are conceived using specific DTD or SCHEMA. To work with CAD or CFD data, it is necessary to define an appropriate data dictionary through which the data will be structured. Benefits of XML include facility of the model to describe any type of data, and in particular, geometrical data and CFD data, human readability, facility of interpretation (using either SAX or DOM parsers[9]) and good integration with the Web.

From STEP to XML

Two approaches have been proposed to migrate data from STEP to XML : Late Binding and Early Binding ([10]). In the case of Late Binding, the names composing the XML vocabulary correspond directly to the data types and the attributes defined in the EXPRESS model. In the case of Early Binding, a specific format is used for each model. Early Binding leads to much more typified and much larger DTD's, but they are easier to interpret for specific models.

The STEP TOOLS ([1]) company developed a library named ST-XML to ease the representation of STEP files in XML. The binding is composed of two stages. In the first stage, starting from the EXPRESS data model, the corresponding data dictionary (DTD) is generated. In the second stage, the corresponding XML file is generated starting from the STEP data file and on the basis of the generated DTD.

CFD data representation using XML

Efforts have also been made to represent CFD data using XML. For instance, Lin and al.[11] have recently proposed a framework infrastructure strictly based on XML for CFD representation. Even though such an approach has great merits from the data representation point of view, it still entails in most cases a translation phase to integrate with commercial solvers, as these packages are currently not outputting their solution in XML format.

Summary

Based on the results of this technological survey (a more thorough version is available in[12]), XML clearly stands as the most promising framework for data exchange. We have thus chosen to steer OPALE project developments in that direction. On the CAD and CFD fronts, though, STEP and CGNS have acquired a significant momentum that should not be neglected. To lessen the risks of this relatively short project (two years in its current phase), we have thus chosen to directly integrate the STEP and CGNS technologies into an XML-based framework.

As a first phase in the integration process, CGNS and STEP were analyzed in terms of their specification model. As mentioned earlier, both standards share a three level specification structure, comprising an abstract concept specification, a manipulation interface specification and a persistent data specification, as illustrated in Fig. 2.

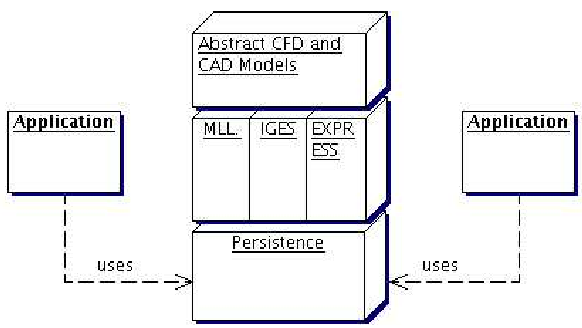

Fig. 2: Three layer standard specification

Comparing this specification structure to the one proposed by XML, one realizes that in the context of XML, the definition of the conceptual level is left to the user, who must specify concepts and their grammar through a DTD or a SCHEMA. As discussed in the next section, the task of defining the concepts for the OPALE project did not simply consist in translating the SIDS specification into an XML DTD, but rather involved many refinements to the conceptual model for the specific context of the application.

3. Proposed Model

Numerical analysis and design processes ask for a robust, complete and easy to understand CFD data model, such as all the modules in the analysis process will be able to query and update it (see Fig. 3). To reach such a flexible CFD data standard matching all CFD needs would be ideal, but would present less guarantees of coherency and applicability. Trying to build a robust CFD data model, conforming with the SIDS ([8]) has in fact raised several CGNS interpretation problems. This has lead to the development of a constrained model, that restricts the way the CGNS system is used. The OPALE model is less generic than the CGNS model, nevertheless it is complete and coherent in the context of hydraulic turbine simulations. Such a specialization is called a standard profile in ([13]). In the cited paper, the authors propose that the SIDS syntax is directly used to define profiles. We rather define the model through a grammar which can also be used for compliance checking. Our model is published on the Web under the form of a DTD ([14]).

Among the ambiguities that needed to be addressed in CGNS, most can be traced to a will of producing a highly general and flexible model. This has introduced in the CGNS model a large number of optional and repeatable informations that may be present or not, while maintaining validity of the model. Nevertheless, in the context of a specific application, such as the simulation and optimization of hydraulic turbines, much of this flexibility may be and needs to be constrained in order to maintain coherency and to allow for comparisons among analysis runs from different solvers.

A number of decisions on the interpretation of the CGNS standard, and further specifications have been developed, that may be classified among the following model characteristics and data categories:

- constraints on multiple instances of data.
- constraints on optional data,
- specification of interface data,
- specification of reference frame,
- specification of descriptors.

Decision in each of these categories will be discussed and exemplified using concrete aspects of the OPALE model.

3.1. Constraints on multiple instances of data

CGNS mentions the possibility of several databases coexisting in a single file, therefore using several nodes of this type in the same file. The CGNSBase node is the highest node in a CGNS database. It makes possible to gather all the entities needed to completely describe a CFD problem. The CGNS standard does not mention anything concerning the way to treat the case of multiple bases in a single file, or on how to interpret each base. Because of the various interpretation problems that this can lead to, multiple bases are not authorized in the OPALE model.

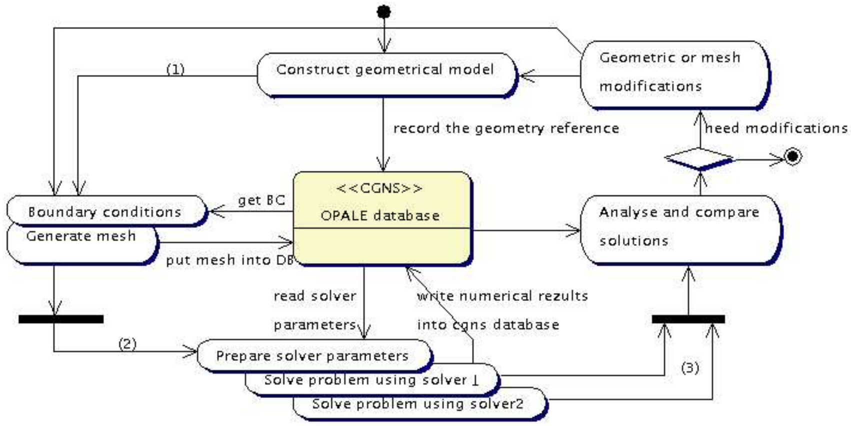

Fig. 3: Data centric analysis process

The same type of discussion also applies to reference state nodes. A single instance of this node is allowed at the base level.

3.2. Constraints on optional data

Numerous fields and child nodes are identified as optional or repeatable in the CGNS data model. Some of these nodes, under most circumstances are nevertheless needed to guarantee the coherency of the data model. In the OPALE model, some of the optional fields are prescribed as specific children of a node, and can only be specified that way.

CGNS has the flexibility to define dimensional units at several levels, complying with the rule of precedence in the hierarchy [see SIDS section 4]. It was decided to impose some restrictions to that rule in the OPALE model, by imposing the existence of globally applicable dimensional units at the base level. That does not mean that the units cannot be redefined at different levels of the data structure using the rule of precedence, but this ensures that a default units system is defined at the highest level of the database.

To be able to describe the DataClass and all other useful dimensional informations to handle data, the DataClass and DimensionalUnits nodes are required under the CGNSBase node. DataClass is an enumeration type that divides data into different categories depending on dimensional units or normalization associated with data. In the OPALE model, the DataClass is fixed to "*Dimensional"* which means that all data are stored with units. Because the data are required to be "*Dimensional"*, the node DimensionalUnits that describes the system of units used to measure the dimensional data is required.

The same type of discussion also applies to data conversion and family nodes. Data conversion nodes are explicitly excluded from the OPALE model, while at least a family node is required at the base level.

Figure 4 illustrates the complete set of children required under the only base node of an OPALE model. As shown in the figure, all nodes are mandatory, but some may be repeated a variable number of time.

3.3. Specification of interface data

In the context of turbo-machinery, two types of interfaces must be specified. The first are periodic interfaces, and the second are rotor/stator (or stage) interfaces.

Currently, CGNS only allows to specify periodic boundary conditions using the most general type of inter-zone interfaces (`GridConnectivity_t`). This is achieved by specifying a “periodic” grid property node as a child of a grid connectivity node. This way disallows the use of a 1 to 1 connectivity, such as would usually be the case in a multi-bloc structured grid. A recent proposal was made to also allow periodic boundary condition specifications under `GridConnectivity1to1_t` nodes, which we strongly support.

The same type of discussion also applies to stage interface (`AverageInterface_t`) nodes.

A third type of interface needs also to be considered. In the post-processing phase of a CGNS file, a number of treatments require to identify specific boundaries or sets of elements on which to perform a given computation. A specific example of such a treatment is the computation of a loss coefficient between the inlet and outlet of the machine, for which the computational domain has been decomposed into multiple zones. In the OPALE model, a decision must be made on how to uniquely identify these regions of interest: boundary conditions imposed on family nodes are the mandatory way of treating such regions.

3.4. Specification of reference frame data

GridCoordinates represents the data structure which describes the physical grid coordinates [see the SIDS, section 7]. The CGNS standard brings the possibility to describe a grid using several coordinate systems (Carte-

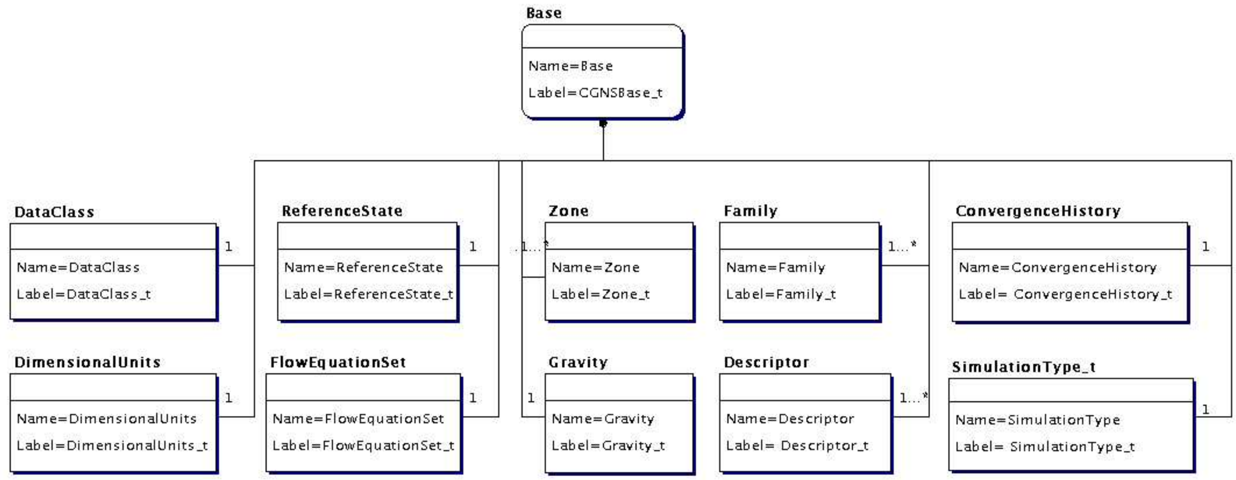

Fig. 4: The CGNSBase node

sian, cylindrical, spherical and auxiliary). Moreover, the standard gives the possibility to specify the dimensional units separately for each coordinate describing the selected system. For example, according to the SIDS, a 3D grid defined using a Cartesian coordinating system (CoordinateX, CoordinateY and CoordinateZ), can have different dimensional units for the three axes. To specify the non-homogeneity it uses the DataClass entity to specify the units system for each axis. If the unit system is the same for the three axes, but differs from the default case, it is enough to define the parameters only once at the level of the GridCoordinates entity. Given the possible confusion and related errors which could bring the use of this flexibility, it was decided to impose that dimensional units on all axes were the same. Moreover, the OPALE model only accepts Cartesian and cylindrical coordinates.

Turbo-machinery being rotating, most computations are performed in relative frames of references. This type of computation is indeed supported by the CGNS data model, but many possible means of specifications are supported under the most general type of specification possible using CGNS.

The `RotatingCoordinates` node is a data structure used to define the center and rotation rate vector of a rotating coordinate system. While this data structure only treats cases where the interfaces between the zones are perpendicular to the axes of rotation, the presence of a `RotatingCoordinates` node in a zone should mean that calculation be made in this revolving reference frame. Logically, one would then expect that the boundary conditions and the solution be specified in the revolving reference frame as well. In particular, the `FlowSolution` node should then have a `RotatingVelocity` child, which does not normally make sense for a `Zone` node not defined with rotating coordinates. This type of coherency is not imposed by the general CGNS model. In the specific context of the OPALE model however, it was decided that flow field and boundary conditions must be homogeneous to the reference frame of the `Zone`, thus allowing only `RotatingVelocity` nodes in the `FlowSolution` of a `Zone` with `RotatingCoordinates`, and stationary velocity and boundary conditions for other `Zones`.

3.5. Specification of descriptor data

The Descriptor data structure is a documentation or annotation structure which allows storing general informations concerning the database. In the CGNS model, Descriptor nodes are always optional, and their content is sometimes suggested, never imposed. We believe that informations concerning the generation of a CGNS file, its author, the solver used, the OPALE file version, etc. must be stored in the database. To this end, and also to keep all the information together, a Descriptor type node has been added as a required child of the base. The kind of information which shall be stored in this node is:

Author #
Date #
InputFile #
Solver #
OpaleFileVersion #

According to the SIDS, the `Descriptor` data structure type can only store character strings, including the special characters nl, tabs, etc. One possibility to store this informations is to use one Descriptor data structure per information, naming the node according to the data stored. The result will be n Descriptor nodes for n parameters to be stored. To clarify and allow flexibility in the data stored by this node, it was chosen to describe it using the XML format. We chose to use the XML data description because of its ability to store the data and its description together. This node can be generated automatically at the time of the optimization task of the CGNS file (please see next section). The XML structure can be defined as follows:

<? xml version="1.0" encoding=" " ?>
<definitionsFile>
<author>

<firstName> FName </firstName>
<lastName> LName </lastName>

</author>
<date> ddmmyyyy </date>
<inputFile> InputFile </inputFile>
<solver> Solver </solver>
<FileVersion> Version </FileVersion>
</definitionsFile>

The same type of discussion also applies to the norm description node which is an optional child of the history convergence node. That node is mandatory in the OPALE model, and its content is structured.

4. Data Standardization using XML

Once the OPALE data model has been decided upon, the next required task to operationalize the model and use it as a central optimization data base, is to allow data from various external sources to be brought into it. To do so in the most flexible and configurable way possible, it was decided to use a fully reconfigurable data import software, that could be tailored to specific characteristics of the various data sources envisioned using XML specification files. As illustrated in Fig. 5, the `XML_CGNS_CONFIG` program can be tailored to the needs of a specific solver through a set of filtering and addition specifications described in XML files.

The data import and transformation process has been divided into three subtasks, based on the characteristics of the CGNS API. These tasks, which will be described next, include :

- Structure extraction
- Structure validation and modification
- Formatting

4.1. CGNS Tree Structure Extraction

Even though entities that form a CGNS database are both conceptually and physically organized in a tree structure, the CGNS Mid-level Library (MLL) does not allow to traverse the tree structure in a convenient and systematic way. As most data manipulation activities can be expressed at the structural (node) level, it was chosen

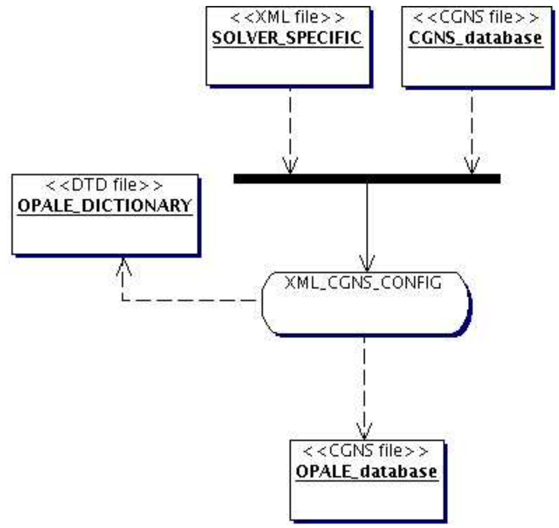

Fig. 5: Using external XML files to format a CGNS data file

to reconstruct the tree structure using low-level file structure (ADF) processing and save this structure for further processing in a XML file. This is the first step in the import process.

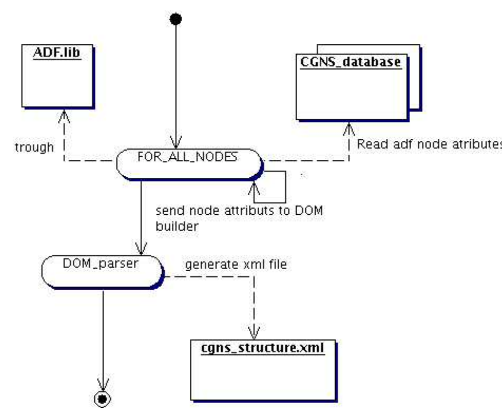

Fig. 6: CGNS Tree Structure Extraction Using ADF Manipulation

Figure 6 illustrates this step, whereby each low level node is traversed to determine its entity content, and the node structure is converted into an XML file using labels identified in accordance with the SIDS and OPALE data models. A DOM parser is used to construct a tree structure from scratch and output it as an XML file.

4.2. Structure Validation and Configuration

The second step gathers the core data manipulation tasks that constitute the standardization process. In, this step, the actual CGNS database structure, previously extracted, is compared to the data model structure prescribed by the OPALE abstract model, and the resulting proposed modifications to the CGNS database are outputted in a configuration file. This mechanism somewhat resembles performing a UNIX *diff* command between the actual CGNS structure and the prescribed structure imposed by the OPALE model. As in the case of *diff*, the validation and configuration step does not alter the CGNS database. It rather constructs a “configuration” file that describes the necessary add and remove operations that must be applied to the database for it to comply with the OPALE model.

To promote extensibility and evolution of the configuration process, an auxiliary data file is provided as input, that specifies a set of standard additions and removals to be performed on a CGNS database. This auxiliary file, which is specifically designed for each supported flow solver, simplifies the standardization process by providing a source for additions to the database, and a list of nodes that always need to be removed.

Taking into account additions and removals specified by the auxiliary data file, the data manipulation task then mostly becomes one of validation. Using a DOM tree structure of CGNS entities, constructed from the extracted XML file and modified through additions and removals specified in the auxiliary file, the standardization process then consists in validating that the final tree structure is valid under the OPALE abstract model expressed as a DTD. This step is schematically illustrated in Fig. 7, and produces another XML file containing the complete sequence of operations that must be performed on the CGNS database.

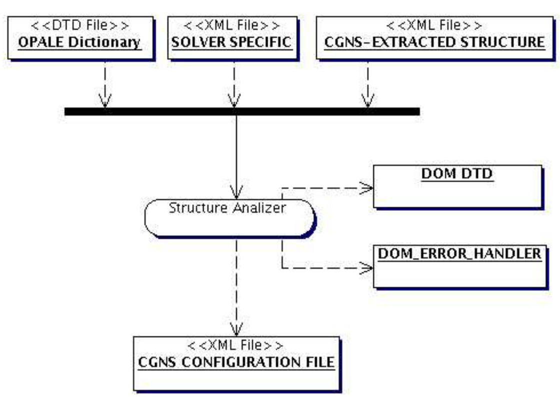

Fig. 7: CGNS Database Validation and Configuration

In the event that the DOM structure, once modified, does not satisfy the OPALE model, an interactive structure manipulation environment is used to fine tune the structure of the database. This tool allows to interactively refine the manipulation process and evolve the auxiliary data file used for a specific type of optimization problem, so that the standardization process be made fully automatic.

4.3. Formating

The third step in the standardization process consists in actually performing the manipulation operations prescribed by the configuration file. In the context of OPALE, these operations are performed using the standard CGNS MLL interface, and can be used to directly modify the original or create a new CGNS database, as shown in Fig. 8.

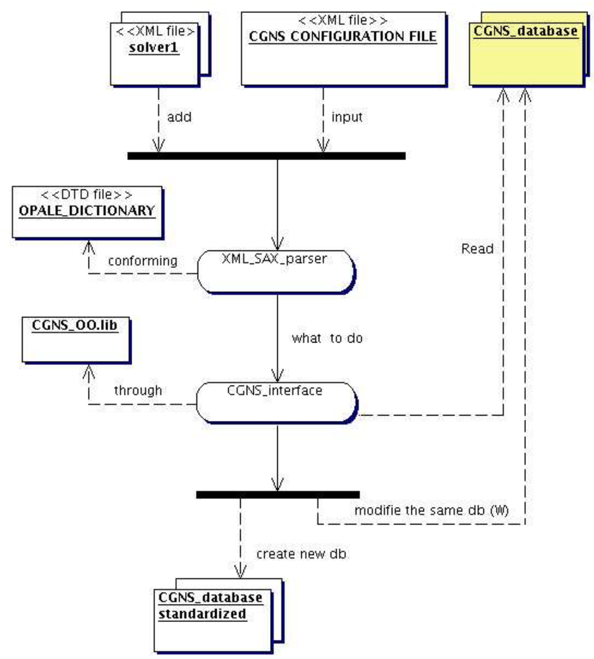

Fig. 8: CGNS Database Formating

This task is performed using a SAX parser which sequentially marches through the node structure and updates the database according to the prescribed operations.

5. Integration for Optimization

The data standardization process defines a database as complete as possible for CFD simulations. Each simulation case is described with all the data necessary to reproduce the flow calculation, as well as the simulation results useful for a complete analysis. This simulation database is supplemented by an optimization database.

The optimization database contains on one hand a description of the optimization problem. This includes a definition of the design variables, a definition of constraints and the objective function. On the other hand, the optimization database stores a set of optimization results consisting in the correspondence between the values of design variables and the value of the cost function (Cf. fig. 9).

The link between the design variables and the blade geometry can be as simple as a restriction to some representation parameters of the blade surface, or much more complex. In realistic cases, some constraints on the thickness of the trailing edge and the surface smoothness, amongst others, must be added in the optimization process. A suitable surface representation and a judicious definition of design variables can reduce the number of constraints which can eased the optimization process.

6. Concluding Remarks

A constrained data model based on the CGNS standard proposal has been presented and discussed. This data model forms the core of an information representation system being developed to address the needs of a turbine blade optimization process. Even though the context of use of the OPALE model is very focused in terms of its application, it is the perception of the authors that most of the discussions, interpretation decisions and comments made in the course of development of the model could readily be applicable for entirely different applications with similar coherency requirements.

First, the definition of the OPALE data model was achieved through a translation and further specification of the core concepts defined by the SIDS, in a DTD grammar usable for XML-based data manipulations. Next, these manipulations have been incorporated into a fully customizable standardization process, guided by the DTD, that could directly interact with the CGNS database. Finally, all process customization and intermediate results were themselves kept in XML files for enhanced evolution and understanding. It is proposed that this standardization approach amounts to introducing a fourth level in the standard model specification approach discussed in § 2, as illustrated in Fig. 10.

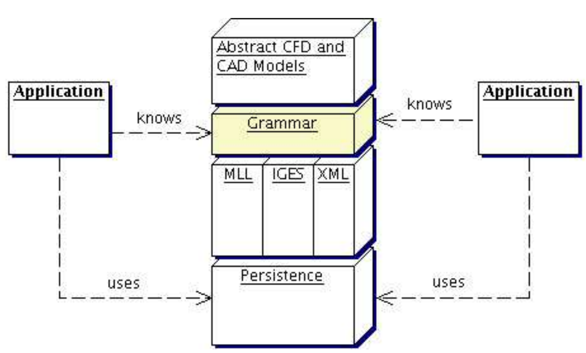

Fig. 10: Four layer standard specification

In this structure, the instantiation of the grammar lies in the DTD specification of the model, which is directly based on concepts introduced in the SIDS. A different application could thus directly benefit from the standardization process simply by implementing a modified grammar (DTD). In future work, migrating to a SCHEMA specification is strongly envisioned. This would allow the construction of hierarchies of related grammars that could be used for standardization in varying application contexts.

Acknowledgments

The authors would like to express their appreciation to the Institut de Recherche d'Hydro-Québec (IREQ) and to the National Science and Engineering Council of Canada (NSERC) for their support.

References

[1] "STEP Tools, Inc." http://www.steptools.com.

[2] "PROSTEP AG Homepage," http://www.prostep.de/en/.

[3] Zhang, Y. P., Zhang, C. C., and Wang, H., "An internet based STEP exchange framework for virtual enterprises," *Computers in Industry*, Vol. 41, No. 4, 2000, pp. 51–63.

[4] P.Bhandarkar, M., Downie, B., and Hardwick, M., "Migrating from IGES to STEP : one to one translation of IGES drawing to STEP drafting data," *Computers in Industry*, Vol. 41, 2000, pp. 261–277.

[5] Basu, D. and Kumar, S., "Importing MESH Entities through IGES/PDES," *Advances in Engineering Software*, Vol. 23, No. 4, 1995, pp. 151–161.

[6] Benyo, T. L., "Project Integration Architecture (PIA) and Computational Analysis Programming Interface (CAPRI) for Accessing Geometry Data from CAD Files," *Proceeding of the 40th AIAA Aerospace Sciences Meeting & Exhibit*, Reno, NV, January 2002.

[7] Poirier, D., Allmaras, S. R., McCarty, D. R., Smith, M. F., and Enomoto, F. Y., "The CGNS System," *AIAA-98-3007*, 1998.

[8] "SIDS documentation," http://www.grc.nasa.gov/WWW/cgns/sids, 2001.

[9] "Extensible Markup Language," http://www.w3.org/XML, 2002.

[10] "XML COVER PAGES, STEP, EXPRESS and XML," http://xml.coverpages.org/stepExpressXML.html.

[11] Lin, R. and Afjeh, A., "A Portable, Self-Validating and Efficient Framework for XML-Based CFD Database Design," *Proceeding of the 41th AIAA Aerospace Sciences Meeting & Exhibit*, 2003, p. 11 pages.

[12] Iepan, H., *Modèle de données pour la représentation d'informations de simulation dans un processus d'optimisation de composantes hydrauliques*, Master's thesis, École Polytechnique de Montréal, 2005.

[13] Poinot, M., Rumsey, C. L., and Mani, M., "Impact of CGNS on CFD Workflow," *34th AIAA Fluid Dynamics Conference & Exhibit*, No. AIAA–2004–2142, June 2004.

[14] "Laboratoire de Maillage et Géométrie Numérique," http://www.polymtl.ca/grmiao/magnu, 2005.

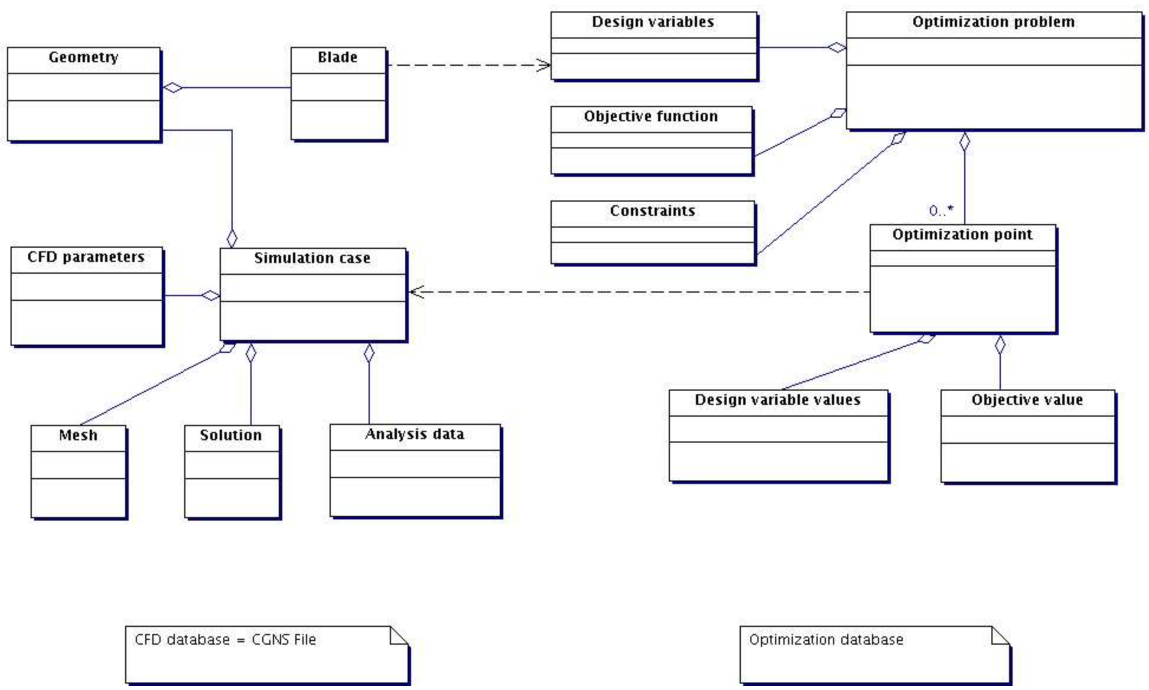

Fig. 9: Links between the CFD database and the optimization database.